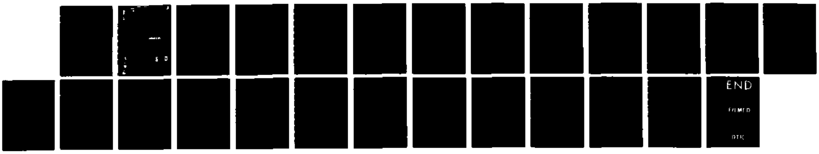
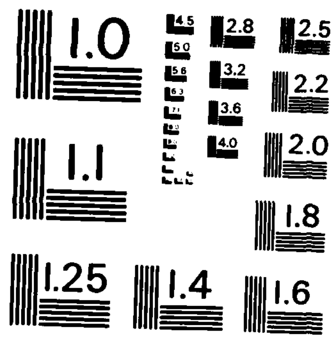
AD-A159 620 KNOWLEDGE ENGINEERING FOR MARINE ARTILLERY CONSULTANT DECISION AID(U) JAYCOR ALEXANDRIA VA MAY 85 N00014-83-C-2045 1/1

UNCLASSIFIED F/G 6/4 NL




1.0
2.8
2.5
3.2
2.2
3.6
4.0
2.0
1.1
1.8
1.25
1.4
1.6

AD-A159 620

DTIC FILE COPY

Knowledge Engineering for the Marine Artillery Consultant Decision Aid

Final Report

May 1985

JAYCOR

205 South Whiting Street
Alexandria, Virginia 22304

2

Knowledge Engineering for the Marine Artillery Consultant Decision Aid

Final Report

May 1985

NRL Contract No. N00014-83-C-2045

JAYCOR No. 6228

Prepared For:

Naval Research Laboratory
Washington, D.C. 20375-5000

Prepared By:

JAYCOR
205 South Whiting Street
Alexandria, Virginia 22304

Table of Contents

1. INTRODUCTION....................................2

2. MAC...4

3. BATTLE..9

4. CONCLUDING REMARKS.............................18

5. BIBLIOGRAPHY...................................19

6. APPENDIX.......................................20

1. INTRODUCTION

This is the final report for the Marine Artillery Consultant project as performed by JAYCOR under contract number N00014-83-C-2045. This work was performed in coordination with personnel from the Navy Center for Applied Research in Artificial Intelligence (NCARAI). This report is a summary of tasks undertaken for the contract including brief descriptions of software written. Software generated under this contract has been transported from a VAX-11/780 to a Sun workstation and successfully tested. All software has been archived on tape and is available at the Center. Some of the software is currently available online in the VAX780 directory "/usr/prj/ramsi".

The need for the work performed under this contract was brought about by the observance that there were a number of tasks being done by Artillery Officers which lent themselves to the use of Artificial Intelligence techniques. In particular, the workload of the Fire Control Officer (FCO) varied widely due to changing "environmental" conditions. This officer receives information on friendly and enemy troop movements and appropriately assigns available artillery units to prioritized targets. These assignments involve not only which artillery units to assign to which target, but also what type of munitions to use and how many volleys to fire. It was felt that, during high load periods, the Fire Control Officer could be assisted by an "intelligent

expert" computer program. This Expert System would help the officer by easing the load, either by presenting condensed information to the officer or by performing certain tasks at a reduced efficiency.

The contract work underwent a number of phases that included the development of a model to encode the knowledge of a human expert, the actual programming of that model, testing and evaluation of the results of using the model, and modification both of the model and the approach used to meet the problem at hand. Throughout the contract period feedback was obtained both from NCARAI staff and Marine Corps personnel on the utility of various aspects of the software. Parts which were negatively received were, in general, modified or deleted, parts which were received with positive remarks were enhanced or expanded. This "feedback and modify" process proved to be extremely beneficial to the evolution of the project.

The following sections describe the evolution of the software produced for the project. This includes descriptions of the original software as well as the modified version of BATTLE that ultimately resulted.

2. MAC

Initially a table driven system was implemented which did indeed respond to "environmental conditions" similar to the responses anticipated from a FCO. The tables were obtained through both interviews with actual experienced Marine Corps personnel and from information gleaned from the effectiveness tables of the Joint Munitions Effectiveness Manual. These tables could be changed during testing to improve the responses given by the system. Factors such as terrain type, target type, target size, ground type (sand, firm, hard rock, etc.) each would influence the ultimate assignments made of weapons to targets.

Various algorithms used for computing effectiveness were implemented during the course of investigation. These algorithms came about through the collaboration of Drs. Lashon Booker, Henry Hamburger, and James Slagle, all NCARAI staff. In essence they each used a complex equation with a number of variables, each variable itself possibly dependent on other variables. Generally the equations heavily used some of the variables as exponentials, a method which made some behavior of the algorithms non-intuitive. The algorithms themselves have been explained in NCARAI reports and will not be covered here (see the Bibliography for further references). Certain side-effects of the switch from one algorithm to another temporarily caused wildly variant answers to be computed. For example, the algorithms have an

equation in them for effectiveness. One piece of information in this equation is an exponent called "relfire" (for "relative fire"). This variable is itself the result of a computation involving a number of other variables, all generally constrained to have ranges somewhat greater than 1 (in general small integers on the order of 10). However, the old MAC algorithm had a number of these variables with ranges from 0 --> 1. During the conversion from old to new, the ranges of these variables were not changed (i.e., table values not updated). The end result was that the "relfire" exponent was always greater than 80 rather than being more reasonably valued. The source code of the MAC system and the tables themselves were updated to reflect the correct equations. During this updating the actual program source code was also restructured to make it easier to read and modify and to improve the modularity (and hence size) of the resulting object code. The user interface was changed to make it harder to break the system while at the same time make it easier for the naive user to use. This involved the modification of the "help" facility both at the top LISP level and within the MAC system itself.

Two sample sets of data (a set of standard artillery pieces and a set of fighting units derived from previous work with the BATTLE system) were then installed. A procedure to allow the automatic inclusion of these sets was written and became an intrinsic part of the MAC system. The entire system was subsequently copied into a location easily

accessible to the Marine expert for his testing and evaluation.

From the comments made by those testing the system and observations made by the implementers of the system it became obvious that any final program with the complexity of MAC would need to have the user interface modified to make it more "user-friendly" and less vulnerable to erroneous input. These modifications involved the elimination of recursion (necessary for those times when a user says "quit" and the program only "pops" out of the current level to the next highest, rather than stopping entirely), greatly improving other aspects of the user interface such as the format of the information presented to the user as well as the possible inputs the user can give, and the installation of a new algorithm (mathematical formula) which would improve in a significant way the results from the system.

In addition to the above standard methods of improving the user interface, the design and installation of a graphics interface was initiated. The graphics interface was designed to allow the MAC system to dynamically display the processes through which the system is going in order to reach a final best recommendation. The graphic output is used to give the uninformed observer a better feel for the kind of processing the MAC system is performing. When the system determines what is considered to be the best recommendation, the display shows the recommendation graphically

as well as textually.

The implementation of the graphics interface required the design of general purpose LISP functions which interacted with a graphics package written in the language C. The output device is a RAMTEK 1024x1280 point full-color display. In general the output consists of a "battle map" with targets and weapons displayed as colored annotated rectangles. Along one side of the display is a legend explaining the graphic representation. Assignments of weapons to targets are displayed by connecting lines of a particular color (denoting shell type). Initial tests made after installation have shown that the graphic results are much easier to appreciate than standard text output (i.e., columns of numbers) by the naive user. This is especially important for future reference if a system is to be designed for use by untrained personnel.

The graphic output of the MAC system was used for a number of purposes other than just aiding the current user of the system. In particular, the output was used to prepare both color viewgraphs and color prints used for presentations and briefings. Over its lifetime the graphic-oriented MAC system was demonstrated to a number of people including NRL Director of Research Dr. T. Coffey, Associate Director of Research Dr. B. Wald, Superintendent of the Information Technology Division Dr. J. Davis, and Dan Leonard (sponsor for the Combat Management (CM) project at the NCARAI). Feed-

back on various aspects of the program were both solicited from and volunteered by the viewers. Finally, the graphic oriented MAC system was demonstrated in September 1983 to representatives from ONR including the Chief of Naval Research Rear Admiral Kollmorgen. It was favorably received.

The design of the new MAC system was continued during this period. This system utilizes an extended expert system which intelligently selects appropriate questions to ask if questioning is necessary. Being a combination of the best of both the old MAC system and the BATTLE system, the new MAC allows much more "intelligent" processing to take place. Additionally, this system has returned to a more intuitive structure for the underlying knowledge base, one which is readily understandable by untrained personnel. The evolution of the newer modified MAC/BATTLE system is explained in the following section.

3. BATTLE

The evolution of the MAC system continued with meetings of the Combat Management project group. These were held to determine an overall structure for the improved MAC/BATTLE expert system. The methodologies used in the PROSPECTOR expert system were discussed at length. JAYCOR personnel obtained and read a journal article on the original thesis work for this system as well as other articles pertaining to the internal structure of the PROSPECTOR system's algorithms and knowledge base. This research included investigation of various methodologies used in data flow analysis as well as examining the possibility of using techniques similar to "relaxation" for certain parts of the underlying mechanisms. Relaxation would be a method whereby a process eventually reaches a decision point after considering (with varying strength) alternate possibilities. Conceptually, this can be envisioned as a "pendulum" coming to rest over time. This iterative method for decisions may prove to be more powerful than a straightforward computation based on "probabilities". The ideas gleaned from this literature survey were used in the writing of a design document for the improved BATTLE system.

A meeting of personnel working both on the Marine Artillery Consultant project and those working on the Multisensor Integration project also took place during the period covered by this report in which the mathematical

theory underlying the MERIT questioning method was discussed. The results of the subject of this meeting had effect on how the merit of potential questions is computed. The implementation of MERIT therefore needed to proceed in general terms until the actual final formulas have been completely determined.

One subject covered by the design document was a draft design for a general purpose resource allocation expert system. This system is to allow multiple problem domains all to be handled by the same core expert system engine. A number of key design issues are to be addressed. They are:

> Inferencing Method -- The inferencing method needs to be robust and general enough to handle multiple problem domains. This implies the complete separation of domain dependencies from the inferencing mechanism implemented. Because the inference engine is the driving force behind an expert system, there should be appropriate structure in both the data and program strategies such that the explanation, questioning, and help facilities can be integrated in a domain independent manner.

> Explanation -- The Resource Allocation expert system will require an extensive explanation facility in order to both aid the user and provide reasonable justifications for particular inferences and decisions made. This facility must be designed in

a generic manner such that diverse domains can be similarly implemented. The generic explanation facility will be an intrinsic part of the inference engine.

Questioning -- The expert system should use a search strategy similar to the MERIT strategy for asking questions of the user. This mechanism should be generalized for situations in which propositions in the inferencing network have more than one consequent.

Help Facility -- At all stages of the expert system execution the user will need the ability to request help. A generic help facility will thus be required which will explain to the user what and why certain information is desired by the system.

User Interface -- The user interface must be as friendly as possible in order to meet the demands to be placed upon the system in actual use. As such, the system must catch erroneous input and either notify the user of the appropriate input required or, in certain cases, correct the information supplied by the user.

With many of these design issues in mind, the BATTLE code was examined to improve the propagation of values through the semantic network intrinsic to its functionality.

The BATTLE system was also enhanced to more fully aid the naive user in answering questions, and more generally, interacting with the program during execution. This included the installation of a HELP mechanism which uses files of text as explanation material when the user requests assistance. The Appendix contains sample output from a session produced by an NRL summer intern. It demonstrates the more informative usage of the system.

Many different copies of the BATTLE source were examined and a master copy, now located in the Resource Allocation/Multisensor Integration project directory, was created. After considerable modification by members of the MAC group this copy became the only official copy kept on the machine.

During this enhancement of the BATTLE system a graphic interface was installed. This interface uses some of the functions previously installed in the MAC system. In addition, other BATTLE-specific functions were written and installed. Both of these required locating the appropriate routines in the BATTLE source code where the graphic display could (but not necessarily would) change given a set of circumstances. Most of the MAC graphics functions used were then modified for the much simpler BATTLE scenario.

One benefit of the modification of the BATTLE system was the knowledge gained of the structure and flow of the program. One concept in particular, the idea of limiting the

time in which to make a decision, was successfully implemented. This alarm mechanism allocates a certain length of time for the network search during the decision process. When the alarm goes off, the most "optimum" solution at that point in the search is given. The graphics output, similar to the MAC system, displays what is considered to be the optimum assignment for the time allocated to the decision. Due to the nature of the BATTLE system, though, multiple weapons are assigned to multiple targets.

Also during this period a menu driven target/target-type selection routine was written and tested on a LMI Lisp Machine. Though this routine (and data structure) is oriented toward the Marine Corps Consultant rather than general resource allocation, the same structure could be used for any menu/submenu selection. This investigation confirmed the ease of use of a "point and pick" menu selection form of input over the original numerically oriented MAC input. Taking full advantage of the powerful menu package available on the Lisp Machine, the user can select from a list of targets, each one displaying a line of explanatory text when the "mouse cursor" is placed over it. When a target is selected (by pushing a mouse button when the cursor is over the item to be selected), another menu "pops up". This menu is a list of relevant target types. For example, if the main target type of "Personnel" is selected, then another menu containing types of "Personnel" ("Working Party", "Infantry", "Bivouac", etc.) pops up. The user then selects one of

these and a unique value is thus returned to the calling process. This allows a very user friendly method for data input, complete with on-line (and dynamic) short explanations of what each item denotes. In addition, the ability to have one of the items be "HELP" for general selection explanation greatly improves the utility of the selection process.

Shortly after this work with the LISP machines was begun, a new JAYCOR researcher was brought on board at the AI Center. He immediately began design work on algorithms and data structures for the automatic acquisition of rule-based knowledge for the project. In addition, he began to use the LISP machines in order to install his software on both these and the VAX machines.

Work also continued on the design and implementation of the knowledge-acquisition rule editor being developed. This editor has been demonstrated and tested by AI Center personnel. The editor allows rules to be input in a much more intuitive manner than had the original BATTLE system. The editor is screen oriented for use on a VT-100 video terminal. Rules are thus displayed appropriately using correct indentation so that the semantics can be much more easily understood. A typical example is given below. This example is very close to the actual way a user would see the rule, thus explicitly demonstrating that ease of use has been greatly enhanced.

IF an active sensor is activated

AND

it's likely that target detected sensor

THEN

target is alerted

The production rule editor being developed was significantly improved through the use of the Flavors style of programming. This method of programming treats objects (such as rules) as entities which can return values after a non-predetermined computation, one which the calling routine does not need to know a priori. This allows a system to be composed of a number of "black boxes", removing the designer from initially having to decide how certain values are obtained. That they are obtained through computation or merely through retrieval is not relevant to the task at hand, hence the designer can concentrate on the overall program structure.

In addition to implementing a rule editor using Flavors, an experimental inferencing mechanism was installed to test some of the ideas for the new BATTLE system. This inferencing mechanism allows the rules not only to be easily created and modified, but also to be tested using sample data. This phase of the rule editor has been demonstrated to others and well received. It was tested more thoroughly and and modified in order to allow a sample P-3 scenario to be implemented.

A taxonomic knowledge acquisition editor for the LMI Lisp Machines underwent initial design and development also during this period. The work was directed towards defining a set of goals and initial implementation of such goals. Primary efforts were to develop a menu driven front end user interface that holds the following properties:

(1) An ability to filter out user requests for creating inconsistent knowledge structures.

(2) An ability to help the user to better understand the structures as he builds them.

(3) An ability to present the user with a set of relational mechanisms for linking together his domain knowledge object descriptions.

(4) An ability to partially define object descriptions and further define them at some later date as domain knowledge is acquired.

(5) An ability to alter object description relationships without the loss of knowledge consistency.

Emphasis has been focused on the development of a general knowledge acquisition tool (KAT). Such a tool will aid a domain expert in arranging his domain knowledge in a natural representation while the system maintains a myriad of data structures that interact to form a global under-

standing of the knowledge.

KAT is intended to be a family of tools for building internal representations of domain knowledge. Initially designed for the MAC/BATTLE scenario, as new domains are studied, it may be necessary to extend KAT's internal structures. Thus many internal structures will remain independant from one another until the knowledge acquisition has progressed to the point where decisions can be made about the types of reasoning that will be necessary for the problem at hand. These structures will then be interrelated in ways that will make inferencing most efficient. Such structures will serve only as intermediate representations of domain knowledge that will be translated into more formal representations with each particular expert system application.

Finally, during the period covered by this report the BATTLE system underwent final transport to one of the visually oriented workstations available at the Center. The functionality available on the VAX-11/780 has been preserved during the transport. Graphics functions were enhanced to allow device-independent use under a CORE graphics-type system.

4. CONCLUDING REMARKS

The CM project changed the focus of its efforts on combining the rigorous mathematical analysis provided by the original MAC program with the capability to allow questioning and explanations provided by the BATTLE system as well as incorporating ideas gleaned from the MACFRONT program, a small test expert system written in OPS5. The design of an inference engine that is smaller and more powerful than the BATTLE inference engine remains a high priority item to date. This inferencing mechanism will be combined with a more natural user interface in a system oriented toward resource allocation, thus allowing other domains (in addition to artillery assignment) to be handled without the usual major program modifications. The Combat Management Project itself underwent reorientation towards general resource allocation. Much of the work performed by JAYCOR under this contract has proven also to be of value for these new directions. It is expected that the enhanced capabilities envisioned for the resource allocation expert system will benefit greatly from experiences gained through the design and implementation of both MAC and BATTLE.

5. BIBLIOGRAPHY

Slagle, J., Gaynor, M., "Expert System Consultation Control Strategy," Proceedings of the National Conference on Artificial Intelligence, Washington, D.C. (August 1983), pp. 369-372.

Hamburger, H., Slagle, J., "Responding to a Call for Fire," NRL Formal Report #3768, Washington, D.C., (December 1983).

Slagle, J., Gaynor, M., Hamburger, H., "A Decision Support System for Fire Support Command and Control," Signal Magazine, Volume 37, #12, (August 1983), pp. 52-60. Published as NRL Report #8769 on December 30, 1983.

Slagle, J., Gaynor, M., "An Intelligent Control Strategy for Computer Consultation," I.E.E.E. Transactions on Pattern Analysis and Machine Intelligence, Volume 6, #2 (March 1984), pp. 129-136.

6. APPENDIX

This appendix contains sample output of the enhanced BATTLE expert system from a session produced by Darin Powers, a summer intern at NRL. This output shows some of the capabilities of the system to aid the user during usage.

The output here has been considerably shortened. Added comments are preceded by the '#' character. User inputs are underlined, all other is output from the BATTLE system.

```
# "af" --- add a friendly unit.

please enter the command or ? for help --- > af

do you want to use the

------ Advanced mode

or

------ Beginner mode

# Use the beginner (naive) mode of input

Enter a or b ---> b

*** ADDING UNITS - BEGINNER MODE ***

please enter the new unit name or s or ? for help -- > ?

** valid unit names can be any string not containing a ?  or
spaces

please enter the next unit name or s or ? -- > fr-1

please enter the unit x position or ? for help -- > ?

** a number is required **

please enter the unit x position or ? for help -- > 100

please enter the unit y position or ? for help -- > 100

please enter a valid friendly unit type or ? -- > ?

# Currently valid friendly unit types are given

VALID ENTRIES INCLUDE:
   1 ) art105 -- mm-105-artillery-unit
   2 ) art155 -- mm-155-artillery-unit
   3 ) art8in -- inch-8-artillery-unit
   4 ) mrt60 -- mm-60-mortar
   5 ) mrt81 -- mm-81-mortar
   6 ) ngf -- naval-gun-fire

please enter a valid friendly unit type or ? -- > 1

please enter the next unit name or s or ? -- > s
```

```
# Now the user desires to give the target.

please enter the command or ? for help --- > at

please enter the next unit name or s or ? -- > tr-1

please enter the unit x position or ? for help -- > 300
please enter the unit y position or ? for help -- > 300
please enter a valid first target component type or ? -- > ?

VALID ENTRIES INCLUDE:
   1 ) art122 -- mm-122-artillery-unit
   2 ) art130 -- mm-130-artillery-unit
   3 ) art152 -- mm-152-artillery-unit
   4 ) fxd -- fixed-target
   5 ) mrl140 -- mm-140-rocket-unit

please enter a valid first target component type or ? -- > 2
please enter the percent of -- art130 > 50
please enter a valid next target component type or ? -- > 3
please enter the percent of -- art152 > 50

please enter the potential target value -- > 65
please enter the next unit name or s or ? -- > s
```

END

FILMED

10-85

DTIC